

Aufgabenblatt 4

(Besprechung am Mo., 12.05.25 und Fr., 16.05.25)

Vorlesung Betriebssysteme und nebenläufige Programmierung Sommersemester 2025

Aufgabe 1: Thread-Zustände und Zustandsübergänge

- Geben Sie die drei wichtigsten Zustände eines Threads an und erläutern Sie sie kurz.
- Geben Sie die möglichen Zustandsübergänge an, sowie jeweils ein Beispiel, wann der Übergang ausgeführt wird.

Aufgabe 2: Thread-Zustandsübergänge

Auf einem Rechner mit einer CPU (d.h., einem Core) gibt es zwei Threads **T1** und **T2**. Thread **T1** ist rechenbereit, Thread **T2** wird gerade ausgeführt. Nun macht **T2** einen Systemaufruf, um auf einen Tastendruck auf der Tastatur zu warten. Welche Zustandsübergänge werden

- beim Eintritt in den BS-Kern bzw. während der Abarbeitung des Systemaufrufs,
- beim Austritt aus dem BS-Kern

ausgeführt? Geben Sie jeweils den Zustandsübergang sowie den Thread an, der den Übergang ausführt!

Aufgabe 3: Elemente von Prozessen und Threads

Welche der folgenden Elemente (bzw. Ressourcen) gehören bei einem Betriebssystem, das mehrere Threads pro Prozess unterstützt, zum Prozeß (sind also allen Threads des Prozesses gemeinsam) und welche sind für jeden Thread privat?

- Offene Dateien
- Tabelle zur Abbildung von logischen auf physische Adressen
- Priorität für den CPU-Scheduler
- Datenregister des Prozessors
- Programmcode
- Befehlszähler
- Kellerzeiger
- Lokale Variable einer Funktion bzw. Methode

Aufgabe 4: Timer-Interrupt

Warum ist es für die ordnungsgemäße Funktion eines Betriebssystems in der Regel erforderlich, daß die Rechnerhardware periodisch (in der Regel alle 10-20ms) einen Interrupt („Timer-Interrupt“) erzeugt?

Aufgabe 5: Threadwechsel

Beschreiben Sie die zwei **wesentlichen** Schritte, die neben der Auswahl des nächsten auszuführenden Threads bei einem Threadwechsel **innerhalb desselben Prozesses** durchgeführt werden müssen.

Anmerkung: dies ist eine Aufgabe im Klausurstil.

Aufgabe 6: Threads mit gemeinsamen Variablen

Laden Sie sich zunächst den Code zum Aufgabenblatt im Archiv [u04Files.zip](#)¹ auf der Vorlesungswebseite herunter und entpacken Sie das Archiv (s. **unzip**).

Das Programm in der Datei **Test.java** erzeugt einen Thread, der eine Referenz auf ein Objekt erhält, das eine ganze Zahl beinhaltet. Der Thread zählt diese Zahl eine Million mal hoch. Übersetzen Sie das Programm und führen Sie es aus, um sich zu vergewissern, dass das ausgedruckte Ergebnis richtig ist.

Erweitern Sie das Programm so, daß ein zweiter Thread erzeugt wird, der eine Referenz auf **dasselbe** Objekt erhält. Warten Sie vor dem Ausdrucken der Zahl, bis auch dieser Thread beendet ist. Die Zahl sollte jetzt also am Ende **zwei** Millionen mal hochgezählt sein. Übersetzen Sie das Programm und führen Sie es aus. Warum stimmt das Ergebnis nicht?

Aufgabe 7: Java Threads

Pflichtaufgabe für die Studienleistung! Abgabe bis So., 18.05., 23:59 über moodle

Implementieren Sie eine Klasse **DruckeZeichen**, die einen Thread realisiert, der n -mal hintereinander einen Buchstaben auf dem Bildschirm ausgibt. Verwenden Sie dabei die Methode **System.out.flush()**, sicherzustellen, daß jede Ausgabe sofort auf dem Bildschirm erscheint. Der auszugebende Buchstabe und die Zahl n sind Eingabeparameter für den Konstruktor und werden als Attribute gespeichert.

Schreiben Sie ein Hauptprogramm, das 8 Threads erzeugt und wieder auf deren Beendigung wartet. Jeder Thread soll dabei einen unterschiedlichen Buchstaben (zwischen 'A' und 'Z') ausgeben, wobei er die Ausgabe 200-mal wiederholt.

Betrachten Sie die Ausgabe Ihres Programms. Wie ändert sich die Ausgabe, wenn Sie nach dem Ausdrucken der Buchstaben eine kurze Verzögerung (z.B. 200 *ms*) einbauen (siehe Methode **Thread.sleep()**) bzw. freiwillig die CPU abgeben (Methode **Thread.yield()**)?

Schreiben Sie Ihre Beobachtung als kurzen Kommentar in Ihren Programmcode!

¹<http://www.bs.informatik.uni-siegen.de/web/wismueller/v1/bs1/u04Files.zip>